



The “Biologically-Inspired Computing” Column

Mike Hinchey*

*NASA Goddard Space Flight Center

1. Editorial: Biologically-Inspired Computing

The field of Biology changed dramatically in 1953, with the determination by Francis Crick and James Dewey Watson of the double helix structure of DNA. This discovery changed Biology for ever, allowing the sequencing of the human genome, and the emergence of a “new Biology” focused on DNA, genes, proteins, data, and search. Computational Biology and Bioinformatics heavily rely on computing to facilitate research into life and development.

Simultaneously, an understanding of the biology of living organisms indicates a parallel with computing systems: molecules in living cells interact, grow, and transform according to the “program” dictated by DNA.

Moreover, paradigms of Computing are emerging based on modelling and developing computer-based systems exploiting ideas that are observed in nature. This includes building into computer systems self-management and self-governance mechanisms that are inspired by the human body’s autonomic nervous system, modelling evolutionary systems analogous to colonies of ants or other insects, and developing highly-efficient and highly-complex distributed systems from large numbers of (often quite simple) largely homogeneous components to reflect the behaviour of flocks of birds, swarms of bees, herds of animals, or schools of fish.

This new field of “Biologically-Inspired Computing”, often known in other incarnations by other names, such as: Autonomic Computing, Pervasive Computing, Organic Computing, Biomimetics, and Artificial Life, amongst others, is poised at the intersection of Computer Science, Engineering, Mathematics, and the Life Sciences. Successes have been reported in the fields of drug discovery, data communications, computer animation, control and command, exploration systems for space, undersea, and harsh environments, to name but a few, and augur much promise for future progress.



This column solicits articles on developments, applications, and case studies of Biologically-Inspired Computing, and any and all of its related areas given above. Please send contributions to: *Dr. Mike Hinchey, Director, NASA Software Engineering Laboratory, Mailstop581.0, NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA*, or via e-mail to: *Michael.G.Hinchey@nasa.gov*

2. Contributed Articles

Biologically Inspired: The Autonomic Computing Initiative

Roy Sterritt* and Michael G. Hinchey**

* University of Ulster, Jordanstown, Northern Ireland

** NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA

Introduction

The human body's autonomic nervous system (ANS) is that part of the nervous system that controls the vegetative functions of the body, such as circulation of the blood, intestinal activity, and secretion and production of chemical “messengers” (hormones) that circulate in the blood. The *sympathetic nervous system (SyNS)* supports “fight or flight”, providing various protection mechanisms to ensure the safety and wellbeing of the body. The *parasympathetic nervous system (PaNS)* supports “rest and digest”, ensuring that the body performs necessary functions for long term health.

Of course, all of this is hidden. While reading this newsletter, the reader is not aware of his/her breathing rate, nor of his/her heart rate. Similarly, if one touches something hot or sharp, one is not aware of the reflex reaction to reconfigure the area in danger to a state that is out of danger—we simply pull back our hand instinctively.

Autonomic Computing

In 2001, IBM launched the Autonomic Computing initiative [1]. At the heart of IBM's vision, is the development of self-managing systems inspired by the human body's ANS, as described above.

The general properties of an autonomic (self-managing) system can be summarised by four objectives: being self-configuring, self-healing, self-optimizing and self-protecting, and four attributes: self-awareness, environment-awareness, self-monitoring and self-adjusting (Figure 1). Essentially, the objectives represent broad system requirements, while the attributes identify basic implementation mechanisms.

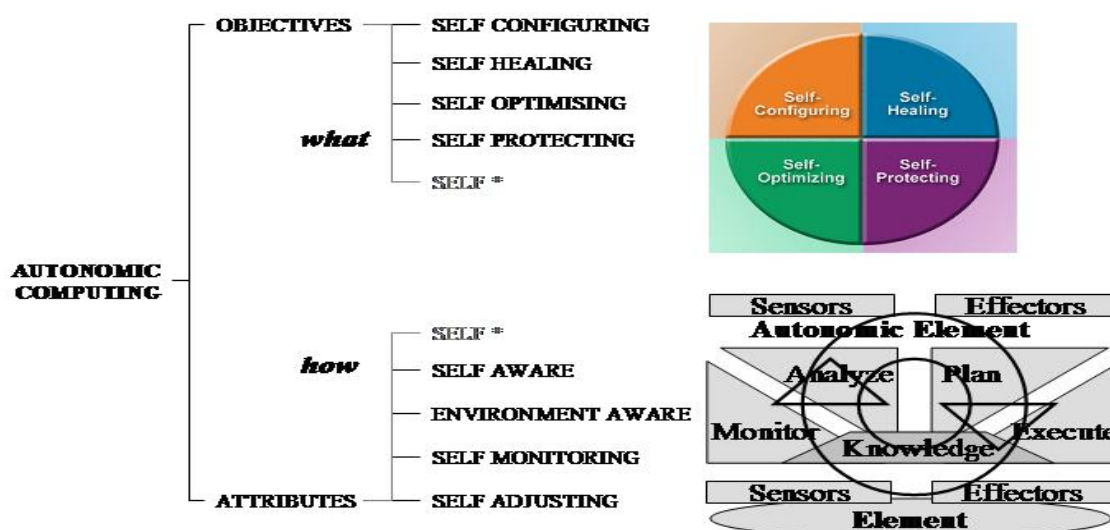


Figure 1: Properties of an Autonomic Computing System.

Self-configuring represents a system's ability to re-adjust itself automatically; this may simply be in support of changing circumstances, or to assist in self-healing, self-optimization or self-protection.

Self-healing, in reactive mode, is a mechanism concerned with ensuring effective recovery when a fault occurs, identifying the fault, and then, where possible, repairing it. In proactive mode, it monitors vital signs in an attempt to predict and avoid "health" problems (reaching undesirable situations).

Self-optimization means that a system is aware of its ideal performance, can measure its current performance against that ideal, and has defined policies for attempting improvements. It may also react to policy changes within the system as indicated by the users. A self-protecting system will defend itself from accidental or malicious external attack. This necessitates awareness of potential threats and a means of handling those threats.

In achieving such self-managing objectives, a system must be aware of its internal state (self-aware) and current external operating conditions (environment-aware). Changing circumstances are detected through self-monitoring and adaptations are made accordingly (self-adjusting). As such, a system must have knowledge of its available resources, its components, their desired performance characteristics, their current status, and the status of inter-connections with other systems, along with rules and policies of how these may be adjusted. Such ability to operate in a heterogeneous environment will require the use of open standards to enable global understanding and communication with other systems [2].

These mechanisms are not independent entities. For instance, if an attack is successful, this will include self-healing actions, and a mix of self-configuration and self-optimisation, in the first instance to ensure dependability and continued operation of the system, and later to increase the self-protection against similar future attacks. Finally, these self-mechanisms should ensure there is minimal disruption to users, avoiding significant delays in processing [7].

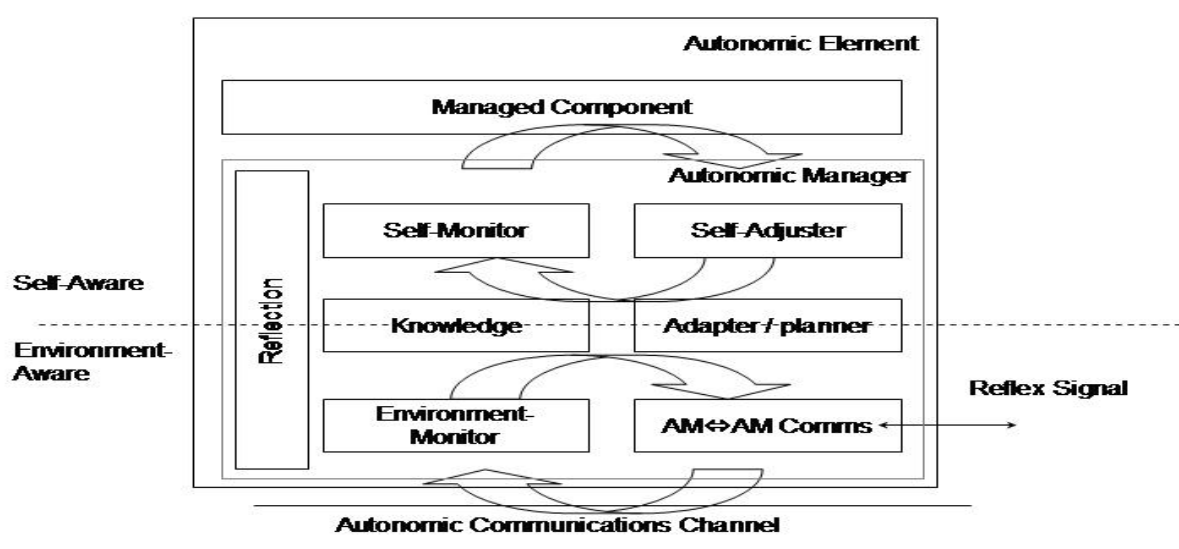


Figure 2: Autonomic Element, with reflection and reflex layers.



Reflex Signals

At the heart of the architecture of any autonomic system are sensors and effectors. A control loop is created by monitoring behavior through sensors, comparing this with expectations (knowledge, as in historical and current data, rules and beliefs), planning what action is necessary (if any), and then executing that action through effectors. The closed loop of feedback control provides the basic backbone structure for each system component [1]. Figure 2 highlights that there are at least two control loops in an Autonomic Element – one for self-awareness and another for environmental awareness [3].

IBM represents this self-monitor/self-adjuster control loop as the monitor, analyze, plan and execute (MAPE) control loop (Figure 1). The monitor-and-analyze parts of the structure process information from the sensors to provide both self-awareness and an awareness of the external environment. The plan-and-execute parts decide on the necessary self-management behavior that will be executed through the effectors. The MAPE components use the correlations, rules, beliefs, expectations, histories, and other information known to the autonomic element, or available to it through the knowledge repository within the AM [1].

The autonomic environment requires that autonomic elements and, in particular, autonomic managers communicate with one another concerning self-* activities, in order to ensure the robustness of the environment. Figure 2 depicts that the autonomic manager communications (AM \leftrightarrow AM) also includes a reflex signal. This may be facilitated through the additional concept of a pulse monitor—PBM (an extension of the embedded system's heart-beat monitor, or HBM, which safeguards vital processes through the emission of a regular “I am alive” signal to another process) with the capability to encode health and urgency signals as a pulse. Together with the standard event messages on the autonomic communications channel, this provides dynamics within autonomic responses and multiple loops of control, such as reflex reactions among the autonomic managers.

This reflex component may be used to safe-guard the autonomic element by communicating its health to another AE. The component may also be utilized to communicate environmental health information. For instance, in the situation where each PC in a LAN is equipped with an autonomic manager, rather than each of the individual PCs monitoring the same environment, a few PCs (likely the least busy machines) may take on this role and alert the others through a change in pulse to indicate changing circumstances [5].

An important aspect concerning the reflex reaction and the pulse monitor is the minimization of data sent – essentially only a “signal” is transmitted. Strictly speaking, this is not mandatory; more information may be sent, yet the additional information must not compromise the reflex reaction. For instance, in the absence of bandwidth concerns, information that can be acted upon quickly and not incur processing delays could be sent.



The important aspect is that the information must be in a form that can be acted upon immediately and not involve processing delays (such as is the case of event correlation).

Just as the beat of the heart has a double beat (lub-dub) the autonomic element's (Figure 2) pulse monitor may have a double beat encoded—as described above, a *self* health/urgency measure and an *environment* health/urgency measure. These match directly with the two control loops within the AE, and the self-awareness and environment awareness properties [8].

Selfware

The initial set of self-* properties, commonly known as self-CHOP (CHOP: configuring, healing, optimizing, protecting), represented the general goal of the Autonomic Computing initiative. They were not intended to be mutually exclusive, nor definitive.

At this stage in this emerging field we are seeking inspiration for new approaches from (obviously, pre-existing) biological mechanisms, and in fact a whole plethora of further self-* properties are being proposed and developed, leading to the coining of the term *selfware*.

Just one such property is *self-destruction*, based on the biological principle of *apoptosis* [4], [7]: If one cuts oneself and starts bleeding, good training results in washing the finger, applying a bandage and carrying on with one's tasks without any further conscious thought. Yet, often, the cut will have caused skin cells to be displaced down into muscle tissue. If they survive and divide, they have the potential to grow into a tumor. The body's solution to dealing with this situation is cell self-destruction (with mounting evidence that cancer is the result of cells not dying fast enough, rather than multiplying out of control, as previously thought).

It is believed that a cell knows when to commit suicide because cells are programmed to do so—self-destruction (sD) is an intrinsic property. This sD is delayed due to the continuous receipt of biochemical retrieves. This process is referred to as *apoptosis*, meaning “drop out”, used by the Greeks to refer to the Autumn dropping of leaves from trees; i.e., loss of cells that ought to die in the midst of the living structure. The process has also been nicknamed “death by default” [4], where cells are prevented from putting an end to themselves due to constant receipt of biochemical “stay alive” signals

Further investigations into the apoptosis process have discovered more details about the self-destruct programme. Whenever a cell divides, it simultaneously receives orders to kill itself. Without a reprieve signal, the cell does indeed self-destruct. It is believed that the reason for this is self-protection, as the most dangerous time for the body is when a cell divides, since if just one of the billions of cells locks into division the result is a tumour, while simultaneously a cell must divide to build and maintain a body.



The suicide and reprieve controls may be compared to the dual-key on a nuclear missile. The key (chemical signal) turns on cell growth but at the same time switches on a sequence that leads to self-destruction. The second key overrides the self-destruct.

Conclusion

Self-managing systems, whether viewed from the perspective of Autonomic Computing, or from that of another initiative, offers a holistic vision for the development and evolution of biologically-inspired computer-based systems. It aims to bring new levels of automation and dependability to systems, while simultaneously hiding their complexity and reducing costs.

A case can certainly be made that *all* computer-based systems should exhibit autonomic properties [6], and we envisage greater interest in, and uptake of, autonomic principles in future system development.

Acknowledgements

Some of the technologies described in this article are patent pending and assigned to the United States government.

References

- [1] Ganek, A.G., Corbi, T.A., The Dawning of the Autonomic Computing Era, *IBM Systems Journal*, 42(1):5–18.
- [2] Horn, P., Autonomic computing: IBM perspective on the state of information technology, IBM T.J. Watson Labs, NY, 15 October 2001. Presented at AGENDA 2001, Scottsdale, AZ (available at <http://www.research.ibm.com/autonomic/>), 2001.
- [3] Sterritt, R., Towards Autonomic Computing: Effective Event Management, In *Proceedings of 27th Annual IEEE/NASA Software Engineering Workshop (SEW)*, Maryland, USA, 3-5 December 2002, IEEE Computer Society, pp 40-47.
- [4] Sterritt, R., Hinchey, M.G., Apoptosis and Self-Destruct: A Contribution to Autonomic Agents?, In *Proceedings of Third NASA-Goddard/IEEE Workshop on Formal Approaches to Agent-Based Systems (FAABS III)*, Greenbelt, MD, 26-27 April 2004, LNAI 3228, Springer-Verlag, pp 262-270.
- [5] Sterritt, R., Autonomic Computing, *Innovations in Systems and Software Engineering: a NASA Journal*, Springer, 1(1), April 2005.
- [6] Sterritt, R., Hinchey, M.G., Why Computer-Based Systems Should be Autonomic, In *Proceedings of the 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005)*, Greenbelt, MD, USA, 3-8 April 2005, pp 406-414.
- [7] Sterritt, R., Hinchey, M.G., Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions, In *Proceedings of IEEE Workshop on the Engineering of Autonomic Systems (EASe 2005) at 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2005)*, Greenbelt, MD, USA, 3-8 April 2005, pp 506-511.
- [8] Sterritt, R., Hinchey, M.G., SPAACE :: Self- Properties for an Autonomous & Autonomic Computing Environment, In *Proceedings of Autonomic & Autonomous Space Exploration Systems (A&A-SES-1) at 2005 International Conference on Software Engineering Research and Practice (SERP'05)*, Las Vegas, NV 27-30 June 2005, CSREA Press, pp 3-8.



3. Forthcoming Related Conferences and Events

EASe 2006

(www.ulster.ac.uk/ease)

EASe 2006, 3rd IEEE Workshop on Engineering Autonomic Systems; held both in Potsdam, Germany, 28-30 March 2006 (co-located with 13th IEEE International Conference on Engineering of Computer Based Systems; <http://ecbs06.hpi.uni-potsdam.de/>) **and** Columbia, Maryland, USA, 24-28 April 2006 (co-located with 2nd IEEE/NASA Systems and Software Week; <http://www.systemsandsoftwareweek.org>).

- *Contact*

Roy Sterritt r.sterritt@ulster.ac.uk

TACS 2006

(cs.okstate.edu/~xiaolin/tacs06/)

TACS 2006, IEEE International Workshop on Trusted and Autonomic Computing Systems, Vienna University of Technology, Vienna, Austria, 18-20 April 2006, co-located with AINA 2006 (<http://www.takilab.k.dendai.ac.jp/conf/aina/2006/>).

- *Contact*

Xiaolin (Andy) Li xlili@caip.rutgers.edu

SEAMS

(www.isr.uci.edu/icse-06/)

SEAMS, ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, Shanghai, China, 21 & 22 May 2006; part of the 28th International Conference on Software Engineering (<http://www.isr.uci.edu/icse-06/>).

- *Important Dates*

1 February 2006	Papers Due
1 March 2006	Notification to Authors
7 March 2006	Camera Ready Due

- *Contact*

Rogério de Lemos r.delemos@kent.ac.uk



BiC 2006

(conferences.computer.org/BiC/)

BiC 2006, 1st IEEE International Conference on Bioinformatics and Complexity, University of Limerick, Ireland, 12 & 13 June 2006

- *Contact*

Vicky Markstein vicky@lifesciencessociety.org

ICAC 2006

(www.autonomic-conference.org)

ICAC 2006, 3rd IEEE International Conference on Autonomic Computing, Dublin, Ireland, 12-16 June 2006

- *Important Dates*

15 January 2006	Abstracts Due
22 January	Full Papers Due
3 March	Notification to Authors
3 April	Final Manuscripts Due

- *Contact*

icac@caip.rutgers.edu

SelfMan 2006

(research.ihost.com/selfman2006/)

SelfMan 2006, 2nd IEEE International Workshop on Self-Managed Networks, Systems and Services, Dublin, Ireland, 16 June 2006; co-located with ICAC 2006 (<http://www.autonomic-conference.org>).

- *Important Dates*

15 February 2006	Papers Due
15 March 2006	Notification of Acceptance
7 April 2006	Camera Ready Due

- *Contact*

selfman2006@gmail.com



DASC 2006

(www.cs.iupui.edu/DASC06/)

DASC 2006, 2nd IEEE International Symposium on Dependable Autonomic and Secure Computing, Indianapolis, Indiana, USA, 29 September-1 October 2006.

- *Important Dates*

15 April 2006

Papers Due

30 May 2006

Notification of Acceptance

1 July 2006

Final Manuscripts Due

- *Contact*

Yuanshun Dai

ydai@cs.iupui.edu

BICC 2006

(www.wcc-2006.org/bicc2006)

BICC 2006, IFIP Conference on Biologically Inspired Cooperative Computing, Santiago, Chile, 20-25 August 2006, as part of IFIP World Computer Congress 2006 (<http://www.wcc-2006.org/>).

- *Important Dates*

15 January 2006

Paper Submission Deadline

15 March 2006

Notification of Acceptance

7 April 2006

Camera ready versions due

- *Contact*

Franz J. Rammig

franz@uni-paderborn.de

Mauricio Solar

msolar@usach.cl

IEEE Task Force on Autonomous and Autonomic Computing (tab.computer.org/AAS/)

IEEE TF on AAS supports and engages in events and activities related to autonomy and autonomic computing. Sign up for membership at: <http://www.computer.org/TCSignup/>

- *Contact*

Roy Sterritt, Task Force Chair

r.sterritt@ulster.ac.uk



THE EASST NEWSLETTER

Life Sciences Society

(www.lifesciencessociety.org)

The Life Sciences Society aims to create a vital organization to enable those engaged in the Life Sciences and related areas to maintain a cutting edge view of current developments in the Life Sciences industry, research and education.

- *Contact*

Vicky Markstein, President

vicky@lifesciencessociety.org